Brultech Research Inc

# GEM Commands, Packet Info and API

3/5/2012

## ***Commands:***

All GreenEye commands are executed beginning with the escape sequence "^^^". Once the GEM receives these three characters, it pauses its normal operation and listens for a valid command sequence. These commands are broken into groups:

- **SYS system**
- **RQS request**
- **Cxx channel**
- **TMP temperature**
- **PLS pulse counter**
- **API data request**

1. System "**SYS**" commands are used to read or modify parameters which apply to the entire system.
2. Request "**RQS**" commands provide the ability to read the current settings.
3. Channel "**Cxx**" commands allow setup of parameters relating to any of the 32(48) power monitoring channels.
4. Temperature "**TMP**" commands are used for setting up the temperature sensor parameters.
5. Pulse "**PLS**" commands are used to setup the pulse counter parameters.
6. API data "**API**" gets monitored data. Typically used for "polling" data when realtime packets are not used.

© Brultech Research Inc

GEM Commands, Packet Info and API ..... 1
Commands: ..... 1
1. System Commands "SYS": ..... 3
2. Channel Commands "Cxx": ..... 12
3. Temperature Sensor Commands "TMP": ..... 14
4. Pulse Counter Commands "PLS": ..... 14
5. Request Information Commands "RQS": ..... 16
6. Data Polling API "API": ..... 20
7. Commands using HTTP GET method: ..... 22

## 1. System Commands “SYS”:

All System Commands begin with the following sequence**: “^^^SYS”** (quotes are not included, cases sensitive)

| Action | Command | Description |
|---|---|---|
| **RealTime “ON”** | “_ON” | Causes the GEM to start sending real-time packets based on the user selected packet format and interval.<br><br>Ex: Sending “^^^SYS_ON”<br>GEM responds with _ON and begins sending the selected packet format after the packet send interval time has elapsed. |
| **RealTime “OFF”** | “OFF” | Causes the GEM to stop sending real-time. If you are sending this command remotely, you should enable the “keep alive” pulse so that the GEM doesn’t lose TCP/IP connection with the server.<br><br>Ex: Sending “^^^SYSOFF”<br>GEM responds with OFF |
| | | |
| | | |
| **Set Packet Format** | “PKT”<xx> | xx = 00 to 99<br><br>The value sent must be two numeric characters long. This means that a value such as “2” should be sent like this:<br>“^^^SYSPKT02”<br>Notice the leading zero padding.<br><br>GEM responds with “PKT” if executed.<br><br>The packet formats are described in the GEM’s Packet Format manual. |
| **Set Packet Send Interval** | “IVL” <xxx> | xxx = 001 to 256<br>This must be three numeric characters long.<br>GEM responds with “IVL” once executed.<br><br>The value represents the number of seconds between each packet send. Realtime must be enabled (_ON) for the packets to be sent. |
| **Set Packet Chunk Size** | “PKS”<val><CR> | <val> is a numeric value between 80 and 65000<br>A carriage return (CR) is required at the end of the command.<br><br>This parameter sets the number of bytes to be sent in each chunk. Since the GEM’s packet size can be fairly large, the packet sometimes need to be broken into small sections (chunks) with slight pauses (milliseconds) between each transmission in order not to surpass the throughput of certain communicating devices. |

| | | |
|---|---|---|
| | | Ex: ^^^PKS240<CR> will cause the packet data to be sent in chunks of 240 bytes.<br><br>The GEM responds with PKS |
| **Set Packet Chunk Interval Time** | "PKI"<val><CR> | <val> is a numeric value between 1 and 65000<br>A carriage return (CR) is required at the end of the command.<br><br>The GEM responds with PKI |
| | | |
| **Set Max Buffer Size** | "BFF"<val><CR> | <val> is a numeric value between 10 and 1700<br>A carriage return (CR) is required at the end of the command.<br><br>The GEM responds with BFF |
| **Enable Wifi Hardware Flow Control Affecting Packet Send Requires RN-171 firmware version 1.44 or greater** | "FL1"<x> | <x> = 0 for disabled or 1 for enabled.<br><br>Make sure RN-171 is set for "set uart flow 1". Option 2 uses the RTS line.<br><br>This option is implemented to prevent buffer issues with the Wifly module.<br><br>The GEM responds with FL1 OR<br>"Hardware Flt" which signifies that the GEM is not configured for Wifi module. |
| **Enable Zigbee Flow Control (COM2).** | "FL2"<x> | <x> = 0 for disabled or 1 for enabled.<br><br>Make sure that XBee flow control is enabled ATD71 (CTS).<br><br>This option applies to realtime packet data transfer from the process buffer only. Commands are issued regardless of flow status.<br><br>This flow control also stops data from being sent to the Wifi module or other COM1 communication module<br><br>The GEM responds with FL2 OR<br>"Hardware Flt" which signifies that the GEM is not configured for XBee module. |
| **PT Type** | "TPT" <xxx> | xxx= 001 to 255<br><br>Value must be three numeric characters long.<br><br>GEM responds with "TPT"<br><br>This value is a constant based on the Potential Transformer's (PT) voltage ratio. A value for this constant is chosen so that the GEM displays the proper AC line voltage. If the GEM displays 116V when a multimeter measures 118V at the power line, this value may be increased slightly to cause the GEM to display 118V.<br>See manual for more information. |

| | | |
|---|---|---|
| **PT Range** | "VRG" <xx> | xx= 00 to 10<br><br>Value must be two numeric characters long.<br><br>GEM responds with "VRG"<br><br>This parameter works in conjunction with the "PT Type" described above. The interpreted voltage is doubled every time the value is reduced by one. For example:<br>Range = 3 Displayed voltage = 117.0 V<br>Range = 2 Displayed voltage = 234.0 V<br><br>Refer to the GEM User's Manual for more details. |
| | | |
| **Reset a Specific Pulse Counter** | "RSP" <x> | x = 1 to 4<br><x> is a numeric value representing the pulse counter channel to be reset.<br>The reset value is zero.<br>Each counter channel is 3-bytes long, therefore can count to 16,777,215 pulses.<br><br>GEM responds with "RSP" if executed. |
| **Reset All Pulse Counters** | "RSTP" | Resets all four pulse channel counters to zero.<br><br>GEM responds with "RSTP" if executed. |
| **Reset All Counters** | "RSTA" | Zeroes all counters:<br>• Channel 1 to 48 watt-second<br>• All pulse counters<br>• Seconds counter<br>GEM responds with "RSTA" after execution. |
| **Reset Seconds Counter** | "RSTS" | Zeroes the seconds incremental counter. This counter increments every second as long as the system is monitoring.<br><br>GEM responds with "RSTS"<br><br>It is used along with watt-second values to calculate Watt. Refer to GEM manual. |
| **Reset the Watt-Second counter for a specified channel.** | "RSTC" <xx> | xx = 01 to 48<br><xx> must be a two character value.<br><br>GEM responds with "RSTC"<br><br>This command zeroes the watt-second counter for the specified channel. Absolute and Polarized counters are both reset. |
| **Set the Line Frequency to 50Hz or 60Hz** | HZ<x> | <x> = "0" (60Hz) OR "1" (50Hz)<br><br>GEM responds with "HZ" |

| | | |
|---|---|---|
| | | |
| | | |
| **Set Date and Time** | "DTM"<yy,mm,dd,hh,mm,ss><CR> | Set the date and time of the GEM. For the most part, this date clock is not required. It is there for future use.<br><br>Ex: "DTM12,08,23,13,30,28"<CR> will set the GEM for August 23rd, 2012 1:30:28 pm<br><br>GEM responds with "DTM" |
| | | |
| | | |
| | | |
| **Bootload COM processor** | | |
| **Bootload Engine processor** | | Wait 500 ms after B87 response |
| | | |
| | | |
| **Set the HTTP post URL** | "URL"<url><CR> | <url> is the URL text such as http://mysite.com<br><br>The last byte must be a CR to indicate the end to the URL data.<br><br>The URL text is limited to 63 characters.<br><br>GEM responds with "URL" |
| **Set the HTTP post URL extension** | "URE"<url ext><CR> | <url ext> is the URL extension text such as /mycode.php<br><br>The last byte must be a CR to indicate the end to the URL extension data.<br>The URL extension text is limited to 63 characters.<br><br>GEM responds with "URE" |
| **Set Data Host Token/Key1** | URT<token><CR> | <token> is the assigned token from the data hosting site. Max 63 characters.<br><br>The last byte must be a CR to indicate the end the token string.<br><br>The token string is limited to 63 characters (not including the CR).<br><br>GEM responds with "URT" |
| **Set Additional Post** | "URK"<x><key> | <x> is the key number from 2 to 8 |

| | | |
|---|---|---|
| **Keys** | <CR> | <key> is the assigned key from the data hosting site. Max 31 characters.<br><br>The last byte must be a CR to indicate the end the token string.<br><br>The token string is limited to 63 characters (not including the CR).<br><br>GEM responds with "URK" |
| | | |
| | | |
| **Set Data Host Device Node** | URN<node><CR> | <node> is the assigned token from the data hosting site. Max 63 characters.<br><br>The last byte must be a CR to indicate the end the token string.<br><br>The token string is limited to 63 characters (not including the CR).<br><br>GEM responds with "URN" |
| **Reset XBee Module** | "ZBR" | |
| **Reset Wifi Module** | "RSW" | |
| **Wifi Auto Reset Interval** | "WAR"<x><CR> | Automatically sends a reset pulse to Wifi module after "x" seconds if data has not been received from COM1 port.<br><br>If x=0 then this feature is disabled<br>Valid Range: 30 to 255 seconds<br><br>If a value between 1 and 29 is entered, the value of 30 will automatically be used.<br><br>The last byte must be a CR to indicate the end.<br><br>This feature has been implemented to deal with issues of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication.<br><br>Refer to GEM manual for more information.<br><br>GEM responds with "WAR" |
| **Wifi No-HTTP-Response Reset Interval** | "WRR"<x><CR> | Automatically sends a reset pulse to Wifi module after "x" number of missing response after sending an HTTP "GET", "POST" or "PUT. Used in cases where the Wifi module may have locked up.<br><br>Valid Range: 2 to 255 missing responses<br><br>The last byte must be a CR to indicate the end.<br><br>This feature has been implemented to deal with issues inherent of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. This method is preferred over the "WAR" method |

| | | |
|---|---|---|
| | | when sending packet data using HTTP GET or PUT method.<br><br>Refer to GEM manual for more information.<br><br>GEM responds with "WRR" |
| **ZigBee Auto Reset Interval** | "ZAR"<x><CR> | Automatically sends a reset pulse to Wifi module after "x" seconds if data has not been received from COM1 port.<br><br>If x=0 then this feature is disabled<br>Valid Range: 30 to 255 seconds<br><br>If a value between 1 and 29 is entered, the value of 30 will automatically be used.<br><br>The last byte must be a CR to indicate the end.<br><br>This feature has been implemented to deal with issues of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication.<br><br>Refer to GEM manual for more information.<br><br>GEM responds with "WAR" |
| **ZigBee No-HTTP-Response Reset Interval** | "ZRR"<x><CR> | Automatically sends a reset pulse to Wifi module after "x" number of missing response after sending an HTTP "GET", "POST" or "PUT. Used in cases where the Wifi module may have locked up.<br><br>Valid Range: 2 to 255 missing responses<br><br>The last byte must be a CR to indicate the end.<br><br>This feature has been implemented to deal with issues inherent of third party hardware. If the communication module locks up or stops responding, this option causes a reset to restart the module, restoring communication. This method is preferred over the "WAR" method when sending packet data using HTTP GET or PUT method.<br><br>Refer to GEM manual for more information.<br><br>GEM responds with "WRR" |
| | | |
| | | |
| **Set Installed Hardware/Module** | "HDW"<x><CR> | No Modules x = 0<br>Wifi only: x = 1<br>Zigbee only x = 2<br>Wifi and Zigbee x = 3<br>Wiznet only COM1 x = 4<br>Wifi and Wiznet (COM2) x = ?<br><br>Informs GEM's processor as to which hardware communication module(s) is installed.<br><br>If this value is not configured, the green system LED will flash every |

| | | |
|---|---|---|
| | | second.<br><br>This setting is important to reduce battery drain during power outages.<br><br>GEM responds with "HDW" |
| **Wifi Baud Reset to 19200** | "WBR" | |
| **Re-Boot COM Processor** | "RBC" | This command restarts the main processor. Used for debugging.<br><br>GEM responds with "RBC" |
| **Re-Boot Engine1 Processor** | "RBE" | Resets the power engine processor. Generally not required.<br><br>GEM responds with "RBE" |
| **Master Polyphase ON/OFF Setting**<br><br>**(New 09/10/12)** | "PH"<x> | X = 0 Polyphase is OFF all phases assume phase A regardless of phase settings.<br>X= 1 Polyphase is ON all channels are assigned the user defined phase setting.<br>X= 2 Three-phase delta with 30degree offset<br>X = 3 Single Phase with CT phase compensation based on CT setting<br><br>GEM responds with "PHx" where "x" is the selected value |
| | | |
| **Enable Channels to be included in the HTTP PUT packet** | "EPC"<xxx>,<xxx>,<xxx>,<xxx>,<xxx>,<xxx><CR> | xxx = 000 to 255<br>Six bit mapped bytes sets which of the 48 channels will be sent with the HTTP post.<br>First value = channels 1 to 8<br>Second value = channels 9 to 16<br>…… and so on<br><br>Ex: EPC007,001,128,000,001,000<CR><br><br>007(1-8): chan 1,2 and 3 are enabled<br>001(8-16): chan 9 enabled<br>128(17-24): chan 24 enabled<br>000(25-32): none enabled<br>001(33-40): chan 33 enabled<br>000(41-48): none enabled |
| **Enable Pulse Counters included in the HTTP PUT packet** | "EPP"<xxx><CR> | xxx = 000 to 015<br>Single bit mapped bytes sets which of the 4 pulse counting channels will be sent with the HTTP post.<br><br>Counter1 = bit0<br>Counter2 = bit1<br>Counter3 = bit3<br>Counter4 = bit4<br><br>Ex: value 004 = CNTR 3 enabled for POST |

| | | |
|---|---|---|
| **Enable Post Temperature** | "EPT"<xxx><CR> | xxx = 000 to 255<br>Single bit mapped bytes sets which of the 8 temperature channels will be sent with the HTTP post.<br><br>Temperature 1 = bit0<br>Temperature 2 = bit1<br>Temperature 3 = bit2<br>Temperature 4 = bit3<br>Temperature 5 = bit4<br>Temperature 6 = bit5<br>Temperature 7 = bit6<br>Temperature 8 = bit7<br><br>Ex: value 017 = T1 and T5 enabled for POST |
| **Set Max Number of Channels To Be Used** | "CMX"<xx><CR> | X = 01 to 48 (two characters)<br>Prevents from processing un-used channels. |
| **Set Number of ThingSpeak Block** | "TSB"<x> | <x> is the number of blocks from 1 to 8<br>Each TS blocks contain 8 channels.<br>Each block has its own unique post "Key". The first key (block1) resides in "Token/Key1". Subsequent keys reside in Key2-8. |
| | | |
| **Read and Store SN from Bootload (Get Serial Number)** | "GSN" | |
| | | |
| | | |
| **Enter Bridge Mode** | "BR1" | Links COM1 directly to COM2 and is baud independent. The serial port or module connected to COM1 has a crossover connection directly to any device on COM2 or COM2's serial port.<br><br>The GEM pauses regular processes while in this mode. The date-time clock is also not functional at this time, therefore if the realtime clock is used for any reason, it should be updated after exiting bridge mode.<br><br>This function is very handy for accessing installed communication devices via a serial port or other communication module. This mode is also useful for upgrading firmware or configuring an XBee module (for example) using COM1 serial port.<br><br>A special sequence is required to exit bridge mode. |
| **Exit Bridge Mode**<br><br>**NOTE: Bridge mode may also be exited by pressing the GEM's PB briefly.** | **NOTE:**<br><br>**Do not send** "^^^" <SYS> | Since the bridge1 mode is baud independent, the command to exit from this mode must also be baud independent. For this reason, a timed sequence of any single character is sent to exit this mode.<br><br>When in bridge mode, the System LED will be ??? . To exit bridge1 mode, send the following sequence: |

| | | |
|---|---|---|
| | Only send the sequence described in the paragraph (right) | <any single character> pause 1.75 second<br><any single character> pause 1.75 second<br><any single character> pause 750 millisecond<br><any single character> pause 750 millisecond<br><any single character> |
| | | |
| | | |
| **Factory Preset** | "FP"<x> | <x> is 0-3<br>0 = no com modules installed<br>1 = Wifi module installed<br>2 = XBee module installed<br>3 = Both modules installed<br>4 = Wiznet Only on COM1<br><br>Presets all default values and sets up the hardware settings including flow control. |
| | | |

2. Channel Commands "Cxx":

**a) Individual Channel Settings** are commands that affect only the specified channel. One command is required for each channel.
**b) All Channel Settings** will set all of the channels to a given value
**c) Group Channel Setting** is used to give each channel with a unique value, however the values for all 48 channels are sent in a structured block to speed up the process.

**c) Individual Channel Settings**: Sets a given channel to a given value.

All individual channel settings begin with the following sequence: "^^^" "Cxx", where "xx" is the channel number 01 to 48.

| Action | Command | Description |
|---|---|---|
| | | |
| **Set channel CT Type** | TYP <xxx> | xxx = 001 to 255 |
| | | |
| **Set channel CT Range** | RNG <x> | x = 0 to 9 |
| | | |
| **Set channel Phase** | PH<x> | x = A or B or C |
| | | |
| **Set channel Phase Compensation**<br><br>**(Added Sept 10,2012)** | PC(x) | X = 0,1,2<br>0 = no phase compensation<br>1 = 4 degrees compensation<br>2 = 8 degrees compensation |
| | | |
| | | |
| **Toggle channel polarity**<br>*(revised Apr 22, 2012)* | "^^^CH"<p><xx> | <p> is "0" or "1" polarity<br><xx> is two character channel number |

**b) All Channel Settings:** Sets all channels to a common setting defined by a single value.

The "All Channel" settings begin with the following sequence: "^^^" "CHE".

| Action | Command | Description |
|---|---|---|
| | | |
| **Set All channel CT Type** | TYP <xxx> | xxx = 001 to 255 |
| | | |
| **Set All channel CT Range** | RNG <x> | x = 0 to 9 |
| | | |
| **Set All channel Phase** | PHx | x = A or B or C |
| | | |
| **Set All Channel Pol 0 (default)** | POL0 | |
| **Set All Channel Pol 1** | POL1 | |
| | | |
| | | |

**c) Group Channel Setting:** Set all channels to an individually defined value. All values are sent in a single block.

The "Group Channel" settings begin with the following sequence: "^^^" "CHG".

| Action | Command | Description |
|---|---|---|
| | | |
| **Set CH1 to CH48 CT Type** | TYP <xxx>,<xxx>, <xxx>, ……. <CR> | xxx = 001 to 255<br>Send 48 comma delimited 3-character values corresponding to the desired CT type for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER" |
| | | |
| **Set All channel CT Range** | RNG <x>,<x>,<x>, …….. <CR> | x = 0 to 9<br>Send 48 comma delimited single character values corresponding to the desired CT range for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER" |
| | | |
| **Set All channel Phase** | PH<x>,<x>,<x>, ……. <CR> | x = A or B or C<br>Send 48 comma delimited single character values corresponding to the desired CT phase for each channel. The carriage return <CR> completes the block. A response is echoed "OK" or "ER" |
| | | |
| | | |
| | | |
| | | |
| | | |

## 3. Temperature Sensor Commands "TMP":

All temperature sensor commands begin with the following sequence: **"^^^" "TMP"** (quotes are not included, cases sensitive)

| Action | Command | Description |
|---|---|---|
| | | |
| **Save Temperature Sensor ROM code** | "ROM <c>, <val1> <val2> <val3> <val4> <val5> <val6> <val7> <val8><CR> | c = temperature channel number "1" to "8"<br>val = three character byte value "000" to "255"<br>val1 = the value after <val> is the byte number 1 to 8 |
| **Enable Temperature Channel** | "EN"<x> | x = Temperature Channel 1 to 8<br>When enabled, temperature is processed and read.<br>NOTE: Leave disabled if not required to reduce required processing. |
| **Disable Temperature Channel** | "DS"<x> | x = Temperature Channel 1 to 8 |
| **REMOVED!<br>Set Resolution (9,10,11 or 12 bit)** | R<x><r> | x = Temperature Channel 1 to 8<br>r = resolution (0,1,2,3 = 9bit,10bit,11bit,12bit) |
| | | |
| **Read then Write ROM code for SINGLE 1-Wire Device connected.** | "ROW"<x> | x = Temperature Channel 1 to 8<br>NOTE: Only one 1-wire device may be connected to the bus when using this command.<br>Reads the ROM code of the connected 1-wire device and saves it to the temperature channel indicated by the value of <x>. |
| **Set Temperature Units (C or F)<br>Sept 12,2012** | "DGC" or "DGF" | DGC sets to Celsius<br>DGF sets to Fahrenheit |
| | | |
| | | |
| | | |

## 4. Pulse Counter Commands "PLS":

All temperature sensor commands begin with the following sequence: **"^^^" "PLS"** (quotes are not included, cases sensitive)

| Action | Command | Description |
|---|---|---|
| | | |
| **Enable Pulse Counter** | "EN"<x> | x = Counter number 1 to 4 |
| **Disable Pulse Counter** | "DS"<x> | x = Counter number 1 to 4 |
| | | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| | | |
| | | |
| | | |

## 5. Request Information Commands “RQS”:

All data Request commands begin with the following sequence: “^^^” “RQS”

| Action | Command | Description |
|---|---|---|
| | | |
| **Read all settings** | “ALL” | Response: “ALL” <CRLF> Then sends Hex values comma delimited:<br>- 1 byte (spare)<br>- 48 bytes channel option<br>- 48 bytes CT types (ch1 to ch48)<br>- 24 bytes CT range (One nibble/channel. Two nibbles/byte.) ex: First CT range byte = 0x23, CH1 CT range = 3 and CH2 CT range = 2<br>- 1 byte = PT type<br>- 1 byte = PT range<br>- 1 byte = packet format selection<br>- 1 byte = packet send interval<br>- 1 byte: Wifi Auto Reset Timer Value (“0” = off)<br>- 1 byte:Wifi Missing Resp Rst Value (“0” = off)<br>- 1 byte: Wifi Options<br>- 1 byte: Xbee Auto Reset Timer Val (“0” = off)<br>- 1 byte: Xbee Missing Resp Rst Val (“0” = off)<br>- 1 byte: XbeeWifi Options<br>- ***1 byte: Primary COM port (not yet implemted)***<br>- 48 bytes = packet options / channel<br>- 1 byte = counter options bits 0 to 3, set for enabled (selects if included in certain packet formats)<br>- 1 byte = temperature enable 8bits, one per temperature channel (selects if included in certain packet formats)<br>- 8 bytes = Temperature 1 ROM code<br>- 8 bytes = Temperature 2 ROM code<br>- 8 bytes = Temperature 3 ROM code<br>- 8 bytes = Temperature 4 ROM code<br>- 8 bytes = Temperature 5 ROM code<br>- 8 bytes = Temperature 6 ROM code<br>- 8 bytes = Temperature 7 ROM code<br>- 8 bytes = Temperature 8 ROM code<br>- 1 byte = Temperature channel enabled to be converted and read.<br>- 1 byte = Additional 1-Wire device options<br>- 1 byte = Hardware Modules Installed (required to reduce battery consumption)<br>- 1 byte = Maximum channels processed.<br>- 48 bytes = reserved for future<br>- 8 bytes = System Settings<br>- 6 bytes = bitwise selects which of the 48 channels sent during an HTTP PUT<br>- 1 byte = bitwise selects which of the 4 pulse counter channels sent during an HTTP PUT .<br>- 1 bytes = bitwise selects which of the 8 temperature channels sent during an HTTP PUT<br>- 2 bytes = General packet options<br>- 1 byte = NET metering options (bi-directional counter? )<br>- 2 bytes = Chunk size<br>- 1 byte = Enabled Pulse counters… which counters are processed.<br>- Others..???? |

| | | |
|---|---|---|
| | | -<br>---------  total ??? (was 374) bytes<br><br>End with <CRLF> |
| **Read all CT Types** | "ATP" | Response: Hex values comma delimited<br><br>Packets sent:<br><br>• 48 comma delimited Hex values<br>• <CRLF> carriage return (ascii 13d,10d) |
| **Read all CT Ranges** | "ARG" | Response: Hex values comma delimited<br><br>Data sent:<br><br>• 24 comma delimited Hex values<br>• <CRLF> carriage return (ascii 13d.10d)<br><br>Each range is represented by a single nibble of each Hex value. The LSB nibble represents the odd channels and MSB nibble the even channels.<br>For example:<br><br>Hex values: 12,32,34,45…….. would represent<br><br>CH1 range = 2<br>CH2 range = 1<br>CH3 range = 2<br>CH4 range = 3<br>CH5 range = 4<br>CH6 range = 3<br>CH7 range = 5<br>CH8 range = 4 |
| **Read All channel Options** | "AOP" | Response: Comma delimited Hex values. One value for each channel.<br><br>The "phase" setting is part of this byte.<br><br>Packets sent:<br><br>• 48 comma delimited Hex values<br>• <CRLF> carriage return (ascii 13d,10d) |
| **Read Channel Phase Setting** | "PH"<xx> | <xx> is 01 to 48 channel value<br>Responds with "A", "B" or "C" |
| **Read Realtime Status** | "RTL" | Responds with "OFF" or "ON" followed by <CRLF> |
| **Read Realtime Send Interval** | "ITV" | Responds with a single hex value representing 0 to 255 seconds followed by <CRLF> |
| **Read Realtime Packet Format** | "RTF" | Responds with a single hex value representing format value followed by <CRLF> |
| | | |
| **Read Temperature ON/Off Status** | "TST" | Responds with "xxxxxxxx" where x = 0 or 1 (off/on) followed by <CRLF><br>Ex: 00010100  temp channel 3 & 5 enabled |

| | | |
|---|---|---|
| **Read Pulse ON/Off Status** | "PST" | Responds with "xxxx" where x = 0 or 1 (off/on) followed by <CRLF><br>Ex: 0100 pulse counter 3 enabled |
| **Read Packet Chunk Size** | "PKS" | Responds with numeric value (two byte size) |
| | | |
| | | |
| **Read Packet Chunk Interval** | "PKI" | |
| **Get Max Buffer Size** | "BFF" | Responds with a value between 10 and 1700.<br>This value is the maximum number of bytes to be buffered and must be set to a value greater than the packet size and chunk size.<br>If set to great, problems may occur trying to transfer too much buffered data should a connection be lost then restored. |
| **Read Specified Channel CT Type and Range** | "Cxx" | xx= 01 to 48 Channel number<br>Responds with two byte hex value representing CT Type and Range values <CRLF> |
| **Read PT Type and Range** | "RPT" | Responds with two byte hex value representing PT Type and Range values <CRLF> |
| **Read Hertz Setting** | "HZ" | Responds with 60Hz or 50Hz |
| **Read 8 Temperature ROM byte setting** | "RTR"<x> | x = 1 to 8<br>Responds with 8 comma delimited hex bytes followed by CRLF |
| **Read Max Channel Setting** | "CMX" | A value of 1 to 48 is returned. This represents the value that the maximum number of channels to be processed is set to. |
| **Master Enable Polarized WS counters for NET metering** | "ENM"<x> | .x = "0" or "1"<br>When set to "1", polarized ws counters are used. If NET metering is not required, set this value to zero.<br>Future updates will provide the ability to select which channel polarized counters are enabled. |
| **Read ThingSpeak number of blocks** | "TSB" | Returns the number of ThingSpeak blocks. Each block contins 8 channels. |
| | | |
| | | |
| **Read URL string** | "URL" | <url string> Followed by CRLF |
| **Read URL extension** | "URE" | <url extension string> Followed by CRLF |
| | | |
| | | |
| **Read Temperature** | "TE"<x> | x is the temperature channel |
| | | |
| **Read Wifi Auto-Reset Value** | "WAR" | Returns Wifi Auto-Reset value |
| **Read Wifi Missing Response Counter Value** | "WRR" | Returns Wifi Missing Response value |
| | | |
| | | |
| | | |
| | | |
| **Read Firmware Version of Master Processor** | "VR1" | Read the firmware version for the communication processor. Ends with CRLF |
| **Read Firmware Version of Engine Processor** | "VR2" | Read the firmware version for CH1 to CH32 engine processor. Ends with CRLF |
| **Read Firmware Version of Expansion Processor (if 16 channel expansion is present)** | "VR3" | Read the firmware version for CH33 to CH48 engine processor. Ends with CRLF |
| | | |

| | | |
|---|---|---|
| **Request time** | "DTM" | Read realtime clock yy-mm-dd hh:mm:ss |
| | | |
| **Read Serial Number** | "SRN" | |
| | | |
| **Read Enabled Post Channels** | "EPC" | Returns:<br><br><xxx>,<xxx>,<xxx>,<xxx>,<xxx>,<xxx><br><br>xxx = 000 to 255<br>Six bit mapped bytes sets which of the 48 channels will be sent with the HTTP post.<br>First value = channels 1 to 8<br>Second value = channels 9 to 16<br>…… and so on<br><br>Ex: EPC007,001,128,000,001,000<CR><br><br>007(1-8): chan 1,2 and 3 are enabled<br>001(8-16): chan 9 enabled<br>128(17-24): chan 24 enabled<br>000(25-32): none enabled<br>001(33-40): chan 33 enabled<br>000(41-48): none enabled |
| **Read Enabled Post Counters** | "EPP" | Returns bit mapped value representing the counters included in HTTP post. |
| **Read Enabled Post Temperature Channels** | "EPT" | Returns bit mapped value representing the temperature channels included in HTTP post. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## 6. Data Polling API "API":

All data Request commands begin with the following sequence: "^^^" "API"

| Action | Command | Description |
|---|---|---|
| | | |
| **Read Single Channel Power (Watt)** | "WAT"<xx> | xx= Channel number 01 to 48<br>Numeric watt data followed by CRLF |
| **Read Power for a Range of Channels (Watt)** | "WRG" <ss><rr> | ss = start channel (01 to 48)<br>rr = number of consecutive channels<br>Numeric Data sent is comma delimited.<br>Ends with CRLF |
| | | |
| **Read Single Channel Energy (WattSecond)** | "ENR"<xx> | xx= Channel number 01 to 48<br>Numeric watt-second data followed by CRLF |
| **Read Absolute for a Range of Channels (WattSeconds)** | "EGR" <ss><rr> | ss = start channel (01 to 48)<br>rr = number of consecutive channels<br>Numeric Watt-Second Data sent is comma delimited.<br>Ends with CRLF |
| **Read Enabled Channels for POST** | "EPC" | Returns six byte values. These bit-mapped values represents which channels are included during a POST<br><br>Returned values are comma delimited.<br><br>Ex: value 5 = counter 1 and counter 3 |
| **Read Enabled Pulse Counters for POST** | "EPP" | Returns single byte value 0 to 15<br>This bit-mapped value represents which pulse counter channels are included during a POST<br><br>Ex: value 5 = counter 1 and counter 3 |
| **Read Enabled Temperature Channels for POST** | "EPT" | Returns single byte value 0 to 255<br>This bit-mapped value represents which temperature channels are included during a POST<br><br>Ex: value 5 = temp 1 and temp 3 |
| | | |
| **Read Voltage** | "VLT" | Sends Numeric Value = Volt x 10.<br>This value should be divided by ten to obtain one decimal point accuracy.<br>Ends with CRLF |
| | | |
| **Read Incremental Seconds Counter** | "SEC" | Send Numeric 3-byte value.<br>Ends with CRLF |
| | | |
| | | |
| | | |
| **Send all recent power, temperature and counter values.**<br><br>**The purpose of this API is to provide updated data to the setup** | "VAL" | Responds with "VAL"<CRLF><br>Sends comma delimited numeric values.<br>c1=<watt>,c2=<watt> …….,t1=<degrees>,t2=<degrees>,……<br>p1=<pulse count lsb value>, …… "END"<CRLF> |

| | | |
|---|---|---|
| **program.** | | |
| | | |
| **Read Temperature of Specified Temperature Channel** | "TP"<x> | "x" is the temperature channel number. |
| | | |
| **Send One Packet** | "SPK" | Sends using the currently selected packet format. Realtime must be OFF in order for this cmd to be functional. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

## 7. Commands using HTTP GET method:

All data Request commands begin with the following sequence: "^^^"

| Action | Command | |
|---|---|---|
| | | |
| **Send All GEM Settings Info** | "RQS" "WEB" | index.php/ecmEngine/gsettings/?SN=&cm=&t1=&t2=&t3=&t4=&t5=&t6=&t7=<br>&t8=&t9=<br>&t10=&t11=&t12=&t13=&t14=&t15=&t16=&t17=&t18=&t19=&t20=&t21=&t22<br>=&t23=<br>&t24=&t25=&t26=&t27=&t28=&t29=&t30=&t31=&t32=&t33=&t34=&t35=&t36<br>=&t37=<br>&t38=&t39=&t40=&t41=&t42=&t43=&t44=&t45=&t46=&t47=&t48=&r1=&r2=<br>&r3=&r4=<br>&r5=&r6=&r7=&r8=&r9=&r10=&r11=&r12=&r13=&r14=&r15=&r16=&r17=&r1<br>8=&r19=<br>&r20=&r21=&r22=&r23=&r24=&r25=&r26=&r27=&r28=&r29=&r30=&r31=&r3<br>2=&r33=<br>&r34=&r35=&r36=&r37=&r38=&r39=&r40=&r41=&r42=&r43=&r44=&r45=&r4<br>6=&r47=<br>&r48=&p1=&p2=&p3=&p4=&p5=&p6=&p7=&p8=&p9=&p10=&p11=&p12=&p1<br>3=&p14=<br>&p15=&p16=&p17=&p18=&p19=&p20=&p21=&p22=&p23=&p24=&p25=&p26<br>=&p27=<br>&p28=&p29=&p30=&p31=&p32=&p33=&p34=&p35=&p36=&p37=&p38=&p39<br>=&p40=<br>&p41=&p42=&p43=&p44=&p45=&p46=&p47=&p48=&pt=&pr=&dp=&si=<br><br>SN = serial number<br>cm = maximum channels used<br>t = channel type<br>r = channel range<br>p = channel phase<br>pt = PT Type<br>pr = PT Range<br>dp = device phase<br>si = send interval |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |